Application of Information Theory, Lecture 12

Accessible Entropy and Statistically Hiding Commitments

Iftach Haitner

Tel Aviv University.

January 20, 2015

Section 1

Commitment Schemes

Motivation

Motivation

- Digital analogue of a safe
- Numerous applications (e.g., zero-knowledge, coin-flipping, secure computations,)

Definition

Definition 1 (Commitment scheme)

An efficient two-stage protocol (S, R).

- Commit stage: The sender S has private input $\sigma \in \{0,1\}^*$ and the common input is 1^n. The commitment stage results in a **joint** output c, the commitment, and a **private** output d of S, the decommitment.
- Reveal stage: S sends the pair (d, σ) to R, and R either accepts or rejects.

Definition

Definition 1 (Commitment scheme)

An efficient two-stage protocol (S, R).

- Commit stage: The sender S has private input $\sigma \in \{0,1\}^*$ and the common input is 1^n. The commitment stage results in a **joint** output c, the commitment, and a **private** output d of S, the decommitment.
- Reveal stage: S sends the pair (d, σ) to R, and R either accepts or rejects.

Completeness: R always accepts in an honest execution.

Definition

Definition 1 (Commitment scheme)

An efficient two-stage protocol (S,R).

- Commit stage: The sender S has private input $\sigma \in \{0,1\}^*$ and the common input is 1^n. The commitment stage results in a **joint** output c, the commitment, and a **private** output d of S, the decommitment.
- Reveal stage: S sends the pair (d,σ) to R, and R either accepts or rejects.

Completeness: R always accepts in an honest execution.

Hiding:. In commit stage: for any R^* and equal length $\sigma,\sigma' \in \{0,1\}^*$, $\Delta^{\mathsf{R}^*}\left((\mathsf{S}(\sigma),\mathsf{R}^*)(1^n),(\mathsf{S}(\sigma),\mathsf{R}^*)(1^n)\right) = \operatorname{neg}(n)$.

Definition

Definition 1 (Commitment scheme)

An efficient two-stage protocol (S, R).

- Commit stage: The sender S has private input $\sigma \in \{0,1\}^*$ and the common input is 1^n. The commitment stage results in a **joint** output c, the commitment, and a **private** output d of S, the decommitment.
- Reveal stage: S sends the pair (d, σ) to R, and R either accepts or rejects.

Completeness: R always accepts in an honest execution.

Hiding:. In commit stage: for any R^* and equal length $\sigma, \sigma' \in \{0,1\}^*$, $\Delta^{\mathsf{R}^*}((\mathsf{S}(\sigma), \mathsf{R}^*)(1^n), (\mathsf{S}(\sigma), \mathsf{R}^*)(1^n)) = \operatorname{neg}(n)$.

Binding: The following happens with negligible prob. for any S^*:

> $\mathsf{S}^*(1^n)$ *interacts with* $\mathsf{R}(1^n)$ *in the commit stage resulting in a commitment* c*. Then* S^* *outputs two pairs* (d, σ) *and* (d', σ') *with* $\sigma \neq \sigma'$ *and* $\mathsf{R}(c, d, \sigma) = \mathsf{R}(c, d', \sigma') = \texttt{Accept}$.

Definition cont.

- Negligible function: $\mu : \mathbb{N} \mapsto \mathbb{N}$ is negligible, if for any $p \in \text{poly}$ $\exists n_p \in \mathbb{N}$ s.t. $\frac{1}{p(n)} < \mu(n)$ for all $n > n_p$.

Definition cont.

- Negligible function: $\mu : \mathbb{N} \mapsto \mathbb{N}$ is negligible, if for any $p \in$ poly $\exists n_p \in \mathbb{N}$ s.t. $\frac{1}{p(n)} < \mu(n)$ for all $n > n_p$.
- Hiding: Perfect, statistical, computational.

Definition cont.

- Negligible function: $\mu : \mathbb{N} \mapsto \mathbb{N}$ is negligible, if for any $p \in$ poly $\exists n_p \in \mathbb{N}$ s.t. $\frac{1}{p(n)} < \mu(n)$ for all $n > n_p$.
- Hiding: Perfect, statistical, computational.
- Binding: Perfect, statistical, computational.

Definition cont.

- Negligible function: $\mu\colon \mathbb{N} \mapsto \mathbb{N}$ is negligible, if for any $p \in \text{poly}$ $\exists n_p \in \mathbb{N}$ s.t. $\frac{1}{p(n)} < \mu(n)$ for all $n > n_p$.
- Hiding: Perfect, statistical, computational.
- Binding: Perfect, statistical, computational.
- Impossible to have simultaneously both properties to be statistical.

Definition cont.

- Negligible function: $\mu\colon \mathbb{N} \mapsto \mathbb{N}$ is negligible, if for any $p \in$ poly $\exists n_p \in \mathbb{N}$ s.t. $\frac{1}{p(n)} < \mu(n)$ for all $n > n_p$.
- Hiding: Perfect, statistical, computational.
- Binding: Perfect, statistical, computational.
- Impossible to have simultaneously both properties to be statistical.
- Suffices to construct “bit commitments"

Definition cont.

- Negligible function: $\mu : \mathbb{N} \mapsto \mathbb{N}$ is negligible, if for any $p \in \text{poly}$ $\exists n_p \in \mathbb{N}$ s.t. $\frac{1}{p(n)} < \mu(n)$ for all $n > n_p$.
- Hiding: Perfect, statistical, computational.
- Binding: Perfect, statistical, computational.
- Impossible to have simultaneously both properties to be statistical.
- Suffices to construct "bit commitments"
- OWFs imply both statistically binding and computationally hiding commitments, and (more difficult) computationally binding and statistically hiding commitments.

Definition cont.

- Negligible function: $\mu: \mathbb{N} \mapsto \mathbb{N}$ is negligible, if for any $p \in$ poly $\exists n_p \in \mathbb{N}$ s.t. $\frac{1}{p(n)} < \mu(n)$ for all $n > n_p$.
- Hiding: Perfect, statistical, computational.
- Binding: Perfect, statistical, computational.
- Impossible to have simultaneously both properties to be statistical.
- Suffices to construct "bit commitments"
- OWFs imply both statistically binding and computationally hiding commitments, and (more difficult) computationally binding and statistically hiding commitments.
- We focus on computationally binding, and statistically hiding commitments (SHC)

Definition cont.

- Negligible function: $\mu : \mathbb{N} \mapsto \mathbb{N}$ is negligible, if for any $p \in \text{poly}$ $\exists n_p \in \mathbb{N}$ s.t. $\frac{1}{p(n)} < \mu(n)$ for all $n > n_p$.
- Hiding: Perfect, statistical, computational.
- Binding: Perfect, statistical, computational.
- Impossible to have simultaneously both properties to be statistical.
- Suffices to construct "bit commitments"
- OWFs imply both statistically binding and computationally hiding commitments, and (more difficult) computationally binding and statistically hiding commitments.
- We focus on computationally binding, and statistically hiding commitments (SHC)
- Canonical decommitment: d is S's coin and c is protocol's transcript of the commit stage, and decomitment verifies consistency.

Definition cont.

- Negligible function: $\mu : \mathbb{N} \mapsto \mathbb{N}$ is negligible, if for any $p \in$ poly $\exists n_p \in \mathbb{N}$ s.t. $\frac{1}{p(n)} < \mu(n)$ for all $n > n_p$.
- Hiding: Perfect, statistical, computational.
- Binding: Perfect, statistical, computational.
- Impossible to have simultaneously both properties to be statistical.
- Suffices to construct “bit commitments"
- OWFs imply both statistically binding and computationally hiding commitments, and (more difficult) computationally binding and statistically hiding commitments.
- We focus on computationally binding, and statistically hiding commitments (SHC)
- Canonical decommitment: d is S's coin and c is protocol's transcript of the commit stage, and decomitment verifies consistency.
- We will focus on constructing the commit algorithm

Section 2

Inaccessible Entropy

Motivation

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A

$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow \mathsf{A}(1^n, h)}}\left[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')\right] = \operatorname{neg}(n)$$

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n : \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A
$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow \mathsf{A}(1^n,h)}}\left[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')\right] = \operatorname{neg}(n)$$

- Implies SHC. (?)

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n : \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A
$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow \mathsf{A}(1^n,h)}} \left[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')\right] = \operatorname{neg}(n)$$

- Implies SHC. (?)

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n : \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A

$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow A(1^n,h)}} \left[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')\right] = \operatorname{neg}(n)$$

- Implies SHC. (?) Believed not to be implied by OWFs.

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A
$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow \mathsf{A}(1^n,h)}}[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')] = \mathrm{neg}(n)$$

- Implies SHC. (?) Believed not to be implied by OWFs.
- Assume for simplicity that $h \in \mathcal{H}_n$ is 2^n to one and that a PPT cannot find a collision in any $h \in \mathcal{H}_n$

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A
$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow \mathsf{A}(1^n, h)}} \left[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')\right] = \operatorname{neg}(n)$$

- Implies SHC. (?) Believed not to be implied by OWFs.
- Assume for simplicity that $h \in \mathcal{H}_n$ is 2^n to one and that a PPT cannot find a collision in any $h \in \mathcal{H}_n$
- Given $h(U_n)$, the (min) entropy of U_n is $n/2$.

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A

$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow \mathsf{A}(1^n,h)}}[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')] = \operatorname{neg}(n)$$

- Implies SHC. (?) Believed not to be implied by OWFs.
- Assume for simplicity that $h \in \mathcal{H}_n$ is 2^n to one and that a PPT cannot find a collision in any $h \in \mathcal{H}_n$
- Given $h(U_n)$, the (min) entropy of U_n is $n/2$.
- Consider PPT A that on input h first outputs y, and then outputs $x \in h^{-1}(y)$ (possibly using additional random coins)

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A

$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow \mathsf{A}(1^n,h)}}\left[x \neq x' \in \{0,1\}^* \land h(x) = h(x')\right] = \operatorname{neg}(n)$$

- Implies SHC. (?) Believed not to be implied by OWFs.
- Assume for simplicity that $h \in \mathcal{H}_n$ is 2^n to one and that a PPT cannot find a collision in any $h \in \mathcal{H}_n$
- Given $h(U_n)$, the (min) entropy of U_n is $n/2$.
- Consider PPT A that on input h first outputs y, and then outputs $x \in h^{-1}(y)$ (possibly using additional random coins)
- What is the entropy of x given h, y and the coins A's used to sample y?

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n : \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A
$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow \mathsf{A}(1^n,h)}}[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')] = \operatorname{neg}(n)$$

- Implies SHC. (?) Believed not to be implied by OWFs.
- Assume for simplicity that $h \in \mathcal{H}_n$ is 2^n to one and that a PPT cannot find a collision in any $h \in \mathcal{H}_n$
- Given $h(U_n)$, the (min) entropy of U_n is $n/2$.
- Consider PPT A that on input h first outputs y, and then outputs $x \in h^{-1}(y)$ (possibly using additional random coins)
- What is the entropy of x given h, y and the coins A's used to sample y?

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A
$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow \mathsf{A}(1^n, h)}} \left[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')\right] = \operatorname{neg}(n)$$

- Implies SHC. (?) Believed not to be implied by OWFs.
- Assume for simplicity that $h \in \mathcal{H}_n$ is 2^n to one and that a PPT cannot find a collision in any $h \in \mathcal{H}_n$
- Given $h(U_n)$, the (min) entropy of U_n is $n/2$.
- Consider PPT A that on input h first outputs y, and then outputs $x \in h^{-1}(y)$ (possibly using additional random coins)
- What is the entropy of x given h, y and the coins A's used to sample y? (essentially) 0!

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n : \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A
$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow A(1^n,h)}}[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')] = \operatorname{neg}(n)$$

- Implies SHC. (?) Believed not to be implied by OWFs.
- Assume for simplicity that $h \in \mathcal{H}_n$ is 2^n to one and that a PPT cannot find a collision in any $h \in \mathcal{H}_n$
- Given $h(U_n)$, the (min) entropy of U_n is $n/2$.
- Consider PPT A that on input h first outputs y, and then outputs $x \in h^{-1}(y)$ (possibly using additional random coins)
- What is the entropy of x given h, y and the coins A's used to sample y? (essentially) 0!
- The 3-block generator $G(h,x) = (h, f(x), x)$ has inaccessible entropy $n/2$

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n : \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A
$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow A(1^n,h)}}\left[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')\right] = \operatorname{neg}(n)$$

- Implies SHC. (?) Believed not to be implied by OWFs.
- Assume for simplicity that $h \in \mathcal{H}_n$ is 2^n to one and that a PPT cannot find a collision in any $h \in \mathcal{H}_n$
- Given $h(U_n)$, the (min) entropy of U_n is $n/2$.
- Consider PPT A that on input h first outputs y, and then outputs $x \in h^{-1}(y)$ (possibly using additional random coins)
- What is the entropy of x given h, y and the coins A's used to sample y? (essentially) 0!
- The 3-block generator $G(h,x) = (h, f(x), x)$ has inaccessible entropy $n/2$
- Does inaccessible entropy generator implies SHC?

Motivation

Definition 2 (collision resistant hash family (CRH))

A function family $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ is collision resistant, if $\forall$ PPT A

$$\Pr_{\substack{h \leftarrow \mathcal{H}_n \\ (x,x') \leftarrow \mathsf{A}(1^n,h)}}[x \neq x' \in \{0,1\}^* \wedge h(x) = h(x')] = \operatorname{neg}(n)$$

- Implies SHC. (?) Believed not to be implied by OWFs.
- Assume for simplicity that $h \in \mathcal{H}_n$ is 2^n to one and that a PPT cannot find a collision in any $h \in \mathcal{H}_n$
- Given $h(U_n)$, the (min) entropy of U_n is $n/2$.
- Consider PPT A that on input h first outputs y, and then outputs $x \in h^{-1}(y)$ (possibly using additional random coins)
- What is the entropy of x given h, y and the coins A's used to sample y? (essentially) 0!
- The 3-block generator $G(h,x) = (h, f(x), x)$ has inaccessible entropy $n/2$
- Does inaccessible entropy generator implies SHC?
- Does OWF implies inaccessible entropy generator?

Real entropy

Real entropy

- Sample entropy: for rv X let $H_X(x) = -\log \Pr_X[x]$.

Real entropy

- Sample entropy: for rv X let $H_X(x) = -\log \Pr_X[x]$.
- $H(X) = \mathrm{E}_{x \leftarrow X}[H_X(x)]$

Real entropy

- Sample entropy: for rv X let $H_X(x) = -\log \Pr_X[x]$.
- $H(X) = \mathrm{E}_{x \leftarrow X}[H_X(x)]$
- Let $G\colon \{0,1\}^n \mapsto (\{0,1\}^\ell)^m$ be an m-block generator and let $(G_1, \ldots, G_m) = G(U_n)$

Real entropy

- Sample entropy: for rv X let $H_X(x) = -\log \Pr_X[x]$.
- $H(X) = \mathrm{E}_{x \leftarrow X}[H_X(x)]$
- Let $G\colon \{0,1\}^n \mapsto (\{0,1\}^\ell)^m$ be an m-block generator and let $(G_1, \ldots, G_m) = G(U_n)$
- For $\mathbf{g} = (g_1, \ldots, g_m) \in \mathrm{Supp}(G_1, \ldots, G_m)$, let

$$\mathrm{RealH}_G(\mathbf{g}) = \sum_{i \in [m]} H_{G_i | G_1, \ldots, G_{i-1}}(g_i | g_1, \ldots, g_{i-1})$$

Real entropy

- Sample entropy: for rv X let $H_X(x) = -\log \Pr_X[x]$.
- $H(X) = \mathrm{E}_{x \leftarrow X}[H_X(x)]$
- Let $G\colon \{0,1\}^n \mapsto (\{0,1\}^\ell)^m$ be an m-block generator and let $(G_1, \ldots, G_m) = G(U_n)$
- For $\mathbf{g} = (g_1, \ldots, g_m) \in \mathrm{Supp}(G_1, \ldots, G_m)$, let

$$\mathrm{RealH}_G(\mathbf{g}) = \sum_{i \in [m]} H_{G_i|G_1,\ldots,G_{i-1}}(g_i|g_1, \ldots, g_{i-1})$$

- The real Shannon entropy of G is $\mathrm{E}_{\mathbf{g} \leftarrow G(U_n)}[\mathrm{RealH}_G(\mathbf{g})]$

Real entropy

- Sample entropy: for rv X let $H_X(x) = -\log \Pr_X[x]$.
- $H(X) = \mathrm{E}_{x \leftarrow X}[H_X(x)]$
- Let $G\colon \{0,1\}^n \mapsto (\{0,1\}^\ell)^m$ be an m-block generator and let $(G_1, \ldots, G_m) = G(U_n)$
- For $\mathbf{g} = (g_1, \ldots, g_m) \in \mathrm{Supp}(G_1, \ldots, G_m)$, let

$$\mathrm{RealH}_G(\mathbf{g}) = \sum_{i \in [m]} H_{G_i | G_1, \ldots, G_{i-1}}(g_i | g_1, \ldots, g_{i-1})$$

- The real Shannon entropy of G is $\mathrm{E}_{\mathbf{g} \leftarrow G(U_n)}[\mathrm{RealH}_G(\mathbf{g})]$
- $\mathrm{E}_{\mathbf{g} \leftarrow G(U_n)}[\mathrm{RealH}_G(\mathbf{g})] = \sum_{i \in [m]} H(G_i | G_1, \ldots, G_{i-1}) = G(U_n)$

Real entropy

- Sample entropy: for rv X let $H_X(x) = -\log \Pr_X[x]$.
- $H(X) = \mathrm{E}_{x \leftarrow X}[H_X(x)]$
- Let $G\colon \{0,1\}^n \mapsto (\{0,1\}^\ell)^m$ be an m-block generator and let $(G_1, \ldots, G_m) = G(U_n)$
- For $\mathbf{g} = (g_1, \ldots, g_m) \in \mathrm{Supp}(G_1, \ldots, G_m)$, let

$$\mathrm{RealH}_G(\mathbf{g}) = \sum_{i \in [m]} H_{G_i|G_1,\ldots,G_{i-1}}(g_i|g_1, \ldots, g_{i-1})$$

- The real Shannon entropy of G is $\mathrm{E}_{\mathbf{g} \leftarrow G(U_n)}[\mathrm{RealH}_G(\mathbf{g})]$
- $\mathrm{E}_{\mathbf{g} \leftarrow G(U_n)}[\mathrm{RealH}_G(\mathbf{g})] = \sum_{i \in [m]} H(G_i|G_1, \ldots, G_{i-1}) = G(U_n)$

- In the actual construction, we sometimes measure the (real) entropy of some of the output blocks.

Accessible entropy

Accessible entropy

- Let G be an m block generator

Accessible entropy

- Let G be an m block generator
- Let $\widetilde{G}$ be an m-block generator, that uses coins r_i before outputting its i'th block (w_i, g_i).

Accessible entropy

- Let G be an m block generator
- Let $\widetilde{G}$ be an m-block generator, that uses coins r_i before outputting its i'th block (w_i, g_i).
- Let $\widetilde{T} = (R_1, W_1, \widetilde{G}_1, \ldots, R_m, W_m, \widetilde{G}_m)$ be the induced rv's in a random execution of $\widetilde{G}$

Accessible entropy

- Let G be an m block generator
- Let $\widetilde{G}$ be an m-block generator, that uses coins r_i before outputting its i'th block (w_i, g_i).
- Let $\widetilde{T} = (R_1, W_1, \widetilde{G}_1, \ldots, R_m, W_m, \widetilde{G}_m)$ be the induced rv's in a random execution of $\widetilde{G}$
- $t = (r_1, w_1, g_1, \ldots, r_m, w_m, g_m) \in \mathrm{Supp}(\widetilde{T})$ is valid with respect to G, if $(g_1, \ldots, g_i) = G(w_i)_{1,\ldots,i}$ for every $i \in [m]$.

Accessible entropy

- Let G be an m block generator
- Let $\widetilde{G}$ be an m-block generator, that uses coins r_i before outputting its i'th block (w_i, g_i).
- Let $\widetilde{T} = (R_1, W_1, \widetilde{G}_1, \ldots, R_m, W_m, \widetilde{G}_m)$ be the induced rv's in a random execution of $\widetilde{G}$
- $t = (r_1, w_1, g_1, \ldots, r_m, w_m, g_m) \in \mathrm{Supp}(\widetilde{T})$ is valid with respect to G, if $(g_1, \ldots, g_i) = G(w_i)_{1,\ldots,i}$ for every $i \in [m]$.
- We will assume for simplicity that the string **t** in consideration is always valid, and omit the w's from the notation.

Accessible entropy

- Let G be an m block generator
- Let $\widetilde{G}$ be an m-block generator, that uses coins r_i before outputting its i'th block (w_i, g_i).
- Let $\widetilde{T} = (R_1, W_1, \widetilde{G}_1, \ldots, R_m, W_m, \widetilde{G}_m)$ be the induced rv's in a random execution of $\widetilde{G}$
- $t = (r_1, w_1, g_1, \ldots, r_m, w_m, g_m) \in \mathrm{Supp}(\widetilde{T})$ is valid with respect to G, if $(g_1, \ldots, g_i) = G(w_i)_{1,\ldots,i}$ for every $i \in [m]$.
- We will assume for simplicity that the string $\mathbf{t}$ in consideration is always valid, and omit the w's from the notation.
- $$\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t}) = \sum_{i \in [m]} H_{\widetilde{G}_i|R_1,\ldots,R_{i-1}}(g_i | r_1, \ldots, r_{i-1})$$

Accessible entropy

- Let G be an m block generator
- Let $\widetilde{G}$ be an m-block generator, that uses coins r_i before outputting its i'th block (w_i, g_i).
- Let $\widetilde{T} = (R_1, W_1, \widetilde{G}_1, \ldots, R_m, W_m, \widetilde{G}_m)$ be the induced rv's in a random execution of $\widetilde{G}$
- $t = (r_1, w_1, g_1, \ldots, r_m, w_m, g_m) \in \mathrm{Supp}(\widetilde{T})$ is valid with respect to G, if $(g_1, \ldots, g_i) = G(w_i)_{1,\ldots,i}$ for every $i \in [m]$.
- We will assume for simplicity that the string **t** in consideration is always valid, and omit the w's from the notation.
- $$\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t}) = \sum_{i\in[m]} H_{\widetilde{G}_i|R_1,\ldots,R_{i-1}}(g_i|r_1, \ldots, r_{i-1})$$
- The accessible entropy of $\widetilde{G}$ (with respect to G) is at most k, if $\Pr_{\mathbf{t}\leftarrow\widetilde{T}}\left[\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t}) > k\right] \leq \mathrm{neg}(n)$.

Accessible entropy

- Let G be an m block generator
- Let $\widetilde{G}$ be an m-block generator, that uses coins r_i before outputting its i'th block (w_i, g_i).
- Let $\widetilde{T} = (R_1, W_1, \widetilde{G}_1, \ldots, R_m, W_m, \widetilde{G}_m)$ be the induced rv's in a random execution of $\widetilde{G}$
- $t = (r_1, w_1, g_1, \ldots, r_m, w_m, g_m) \in \mathrm{Supp}(\widetilde{T})$ is valid with respect to G, if $(g_1, \ldots, g_i) = G(w_i)_{1,\ldots,i}$ for every $i \in [m]$.
- We will assume for simplicity that the string **t** in consideration is always valid, and omit the w's from the notation.
- $$\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t}) = \sum_{i\in[m]} H_{\widetilde{G}_i|R_1,\ldots,R_{i-1}}(g_i|r_1, \ldots, r_{i-1})$$
- The accessible entropy of $\widetilde{G}$ (with respect to G) is at most k, if $\Pr_{\mathbf{t}\leftarrow\widetilde{T}}\left[\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t}) > k\right] \leq \mathrm{neg}(n)$.

Accessible entropy

- Let G be an m block generator
- Let $\widetilde{G}$ be an m-block generator, that uses coins r_i before outputting its i'th block (w_i, g_i).
- Let $\widetilde{T} = (R_1, W_1, \widetilde{G}_1, \ldots, R_m, W_m, \widetilde{G}_m)$ be the induced rv's in a random execution of $\widetilde{G}$
- $t = (r_1, w_1, g_1, \ldots, r_m, w_m, g_m) \in \mathrm{Supp}(\widetilde{T})$ is valid with respect to G, if $(g_1, \ldots, g_i) = G(w_i)_{1,\ldots,i}$ for every $i \in [m]$.
- We will assume for simplicity that the string **t** in consideration is always valid, and omit the w's from the notation.
- $$\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t}) = \sum_{i \in [m]} H_{\widetilde{G}_i | R_1,\ldots,R_{i-1}}(g_i | r_1, \ldots, r_{i-1})$$
- The accessible entropy of $\widetilde{G}$ (with respect to G) is at most k, if $\Pr_{\mathbf{t} \leftarrow \widetilde{T}}\left[\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t}) > k\right] \le \mathrm{neg}(n)$. Why not $\mathrm{E}_{\mathbf{t} \leftarrow \widetilde{T}}\left[\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t})\right]$?

Accessible entropy

- Let G be an m block generator
- Let $\widetilde{G}$ be an m-block generator, that uses coins r_i before outputting its i'th block (w_i, g_i).
- Let $\widetilde{T} = (R_1, W_1, \widetilde{G}_1, \ldots, R_m, W_m, \widetilde{G}_m)$ be the induced rv's in a random execution of $\widetilde{G}$
- $t = (r_1, w_1, g_1, \ldots, r_m, w_m, g_m) \in \mathrm{Supp}(\widetilde{T})$ is valid with respect to G, if $(g_1, \ldots, g_i) = G(w_i)_{1,\ldots,i}$ for every $i \in [m]$.
- We will assume for simplicity that the string **t** in consideration is always valid, and omit the w's from the notation.
- $$\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t}) = \sum_{i\in[m]} H_{\widetilde{G}_i|R_1,\ldots,R_{i-1}}(g_i|r_1, \ldots, r_{i-1})$$
- The accessible entropy of $\widetilde{G}$ (with respect to G) is at most k, if $\Pr_{\mathbf{t}\leftarrow\widetilde{T}}\left[\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t}) > k\right] \leq \mathrm{neg}(n)$. Why not $\mathrm{E}_{\mathbf{t}\leftarrow\widetilde{T}}\left[\mathrm{AccH}_{\mathrm{G},\widetilde{G}}(\mathbf{t})\right]$?
- G has inaccessible entropy d, if the accessible entropy of any PPT $\widetilde{G}$ is smaller be at least d than its real entropy

Example

Example

- Let $\mathcal{H} = \{\mathcal{H}_n : \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ be 2^n-to one collision resistant, and assume for simplicity that a PPT cannot find a collision for any $h \in \mathcal{H}_n$.

Example

- Let $\mathcal{H} = \{\mathcal{H}_n : \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ be 2^n-to one collision resistant, and assume for simplicity that a PPT cannot find a collision for any $h \in \mathcal{H}_n$.
- Let G be the 3-block generator $G(h,x) = (h, h(x), x)$

Example

- Let $\mathcal{H} = \{\mathcal{H}_n : \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ be 2^n-to one collision resistant, and assume for simplicity that a PPT cannot find a collision for any $h \in \mathcal{H}_n$.
- Let G be the 3-block generator $G(h,x) = (h, h(x), x)$
- Real entropy of G is $\log|\mathcal{H}_n| + n$

Example

- Let $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^{2n} \mapsto \{0,1\}^n\}$ be 2^n-to one collision resistant, and assume for simplicity that a PPT cannot find a collision for any $h \in \mathcal{H}_n$.
- Let G be the 3-block generator $G(h,x) = (h, h(x), x)$
- Real entropy of G is $\log|\mathcal{H}_n| + n$
- Accessible entropy of G is $\log|\mathcal{H}_n| + \frac{n}{2}$

Section 3

Manipulating Inaccessible Entropy

Entropy equalization

Entropy equalization

Let G be m-bit generator.

Entropy equalization

Let G be m-bit generator.

For $\ell \in \text{poly}$ let $G^{\otimes \ell}$ be the following $\ell - 1 \cdot m$-bit generator

$$G^{\otimes \ell}(x_1, \ldots, x_\ell, i) = G(x_1)_i, \ldots, G(x_1)_m, \ldots, G(x_\ell)_1, \ldots, G(x_\ell)_{i-1}$$

Entropy equalization

Let G be m-bit generator.

For $\ell \in$ poly let $G^{\otimes \ell}$ be the following $\ell - 1 \cdot m$-bit generator

$$G^{\otimes \ell}(x_1, \ldots, x_\ell, i) = G(x_1)_i, \ldots, G(x_1)_m, \ldots, G(x_\ell)_1, \ldots, G(x_\ell)_{i-1}$$

- Assume the accessible entropy of G is (at most) k_A, then $k_A^{\otimes \ell}$, the accessible entropy of $G^{\otimes \ell}$, is at most $k(\ell - 2) + m$.

Entropy equalization

Let G be m-bit generator.

For $\ell \in$ poly let $G^{\otimes \ell}$ be the following $\ell - 1 \cdot m$-bit generator

$$G^{\otimes \ell}(x_1, \ldots, x_\ell, i) = G(x_1)_i, \ldots, G(x_1)_m, \ldots, G(x_\ell)_1, \ldots, G(x_\ell)_{i-1}$$

- Assume the accessible entropy of G is (at most) k_A, then $k_A^{\otimes \ell}$, the accessible entropy of $G^{\otimes \ell}$, is at most $k(\ell - 2) + m$.
- Assume the real entropy of G is k_R, then

Entropy equalization

Let G be m-bit generator.

For $\ell \in$ poly let $G^{\otimes \ell}$ be the following $\ell - 1 \cdot m$-bit generator

$$G^{\otimes \ell}(x_1, \ldots, x_\ell, i) = G(x_1)_i, \ldots, G(x_1)_m, \ldots, G(x_\ell)_1, \ldots, G(x_\ell)_{i-1}$$

- Assume the accessible entropy of G is (at most) k_A, then $k_A^{\otimes \ell}$, the accessible entropy of $G^{\otimes \ell}$, is at most $k(\ell - 2) + m$.
- Assume the real entropy of G is k_R, then
  1. $k_R^{\otimes \ell}$, the real entropy of $G^{\otimes \ell}$, is at least $k_R^{\otimes \ell} = (\ell - 1)K_R$

Entropy equalization

Let G be m-bit generator.

For $\ell \in \text{poly}$ let $G^{\otimes \ell}$ be the following $\ell - 1 \cdot m$-bit generator

$$G^{\otimes \ell}(x_1, \ldots, x_\ell, i) = G(x_1)_i, \ldots, G(x_1)_m, \ldots, G(x_\ell)_1, \ldots, G(x_\ell)_{i-1}$$

- Assume the accessible entropy of G is (at most) k_A, then $k_A^{\otimes \ell}$, the accessible entropy of $G^{\otimes \ell}$, is at most $k(\ell - 2) + m$.
- Assume the real entropy of G is k_R, then
  1. $k_R^{\otimes \ell}$, the real entropy of $G^{\otimes \ell}$, is at least $k_R^{\otimes \ell} = (\ell - 1)K_R$
  2. For any $i \in [(\ell - 1) \cdot m]$ and $(g_1, \ldots, g_{i-1}) \in \text{Supp}(G_1^{\otimes \ell}, \ldots, G_{i-1}^{\otimes \ell})$: $H(G_i^{\otimes \ell} | G_1^{\otimes \ell}, \ldots, G_{i-1}^{\otimes \ell}) = k/\ell$

Entropy equalization

Let G be m-bit generator.

For $\ell \in$ poly let $G^{\otimes \ell}$ be the following $\ell - 1 \cdot m$-bit generator

$$G^{\otimes \ell}(x_1, \ldots, x_\ell, i) = G(x_1)_i, \ldots, G(x_1)_m, \ldots, G(x_\ell)_1, \ldots, G(x_\ell)_{i-1}$$

- Assume the accessible entropy of G is (at most) k_A, then $k_A^{\otimes \ell}$, the accessible entropy of $G^{\otimes \ell}$, is at most $k(\ell - 2) + m$.
- Assume the real entropy of G is k_R, then
  1. $k_R^{\otimes \ell}$, the real entropy of $G^{\otimes \ell}$, is at least $k_R^{\otimes \ell} = (\ell - 1) K_R$
  2. For any $i \in [(\ell - 1) \cdot m]$ and $(g_1, \ldots, g_{i-1}) \in \text{Supp}(G_1^{\otimes \ell}, \ldots, G_{i-1}^{\otimes \ell})$: $H(G_i^{\otimes \ell} | G_1^{\otimes \ell}, \ldots, G_{i-1}^{\otimes \ell}) = k/\ell$
- Assume $k_R \geq k_A + 1$, then for $\ell = m + 2$, it holds that $k_R^{\otimes \ell} \geq k_A^{\otimes \ell} + 1$

Gap amplification and conversion to min entropy

Gap amplification and conversion to min entropy

Let G be an m-block generator and for $\ell \in \mathsf{poly}$, let G^{ℓ} be the ℓ-fold parallel repetition of G.

Gap amplification and conversion to min entropy

Let G be an m-block generator and for $\ell \in \text{poly}$, let G^ℓ be the ℓ-fold parallel repetition of G.

- Assume accessible entropy of G is (at most) k_A, then the accessible entropy of G is at most $k_A^\ell = \ell \cdot k_A$.

Gap amplification and conversion to min entropy

Let G be an m-block generator and for $\ell \in \text{poly}$, let G^ℓ be the ℓ-fold parallel repetition of G.

- Assume accessible entropy of G is (at most) k_A, then the accessible entropy of G is at most $k_A^\ell = \ell \cdot k_A$.
- Assume $H(G_i|G_1, \ldots, G_{i-1}) = k_R$ for any $i \in [m]$, then for any $i \in [m]$ and $(g_1^\ell, \ldots, g_{i-1}^\ell) \in \text{Supp}(G_1^\ell, \ldots, G_{i-1}^\ell)$:
  $k_{min}^\ell = \mathrm{H}_\infty(G_i^\ell|G_1^\ell, \ldots, G_{i-1}^\ell) \approx \ell k_R$

Gap amplification and conversion to min entropy

Let G be an m-block generator and for $\ell \in \text{poly}$, let G^ℓ be the ℓ-fold parallel repetition of G.

- Assume accessible entropy of G is (at most) k_A, then the accessible entropy of G is at most $k_A^\ell = \ell \cdot k_A$.
- Assume $H(G_i | G_1, \ldots, G_{i-1}) = k_R$ for any $i \in [m]$, then for any $i \in [m]$ and $(g_1^\ell, \ldots, g_{i-1}^\ell) \in \text{Supp}(G_1^\ell, \ldots, G_{i-1}^\ell)$:
  $k_{min}^\ell = \text{H}_\infty(G_i^\ell | G_1^\ell, \ldots, G_{i-1}^\ell) \approx \ell k_R$
- If $k_A \le k_R - 1$, then $\forall n \in \text{poly}\ \exists \ell \in \text{poly}$ such that $\ell k_{min}^\ell > k_A^\ell + n$

Section 4

Inaccessible Entropy from OWF

The generator

The generator

Definition 3

Given a function $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let G be the $(n+1)$-block generator $f(x)_1, \ldots, f(x)_n, x$.

The generator

Definition 3

Given a function $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let G be the $(n+1)$-block generator $f(x)_1, \ldots, f(x)_n, x$.

Lemma 4

Assume that f is a OWF then G has accessible entropy at most $n - \log n$.

The generator

Definition 3

Given a function $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let G be the $(n+1)$-block generator $f(x)_1, \ldots, f(x)_n, x$.

Lemma 4

Assume that f is a OWF then G has accessible entropy at most $n - \log n$.

- Recall f is OWF if
  $\Pr_{x \leftarrow \{0,1\}^n; y = f(x)} \left[\mathsf{Inv}(y) \in f^{-1}(y) \right] = \mathsf{neg}(n)$ for any PPT Inv.

The generator

Definition 3

Given a function $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let G be the $(n+1)$-block generator $f(x)_1, \ldots, f(x)_n, x$.

Lemma 4

Assume that f is a OWF then G has accessible entropy at most $n - \log n$.

- Recall f is OWF if
  $\Pr_{x \leftarrow \{0,1\}^n; y=f(x)}\left[\mathrm{Inv}(y) \in f^{-1}(y)\right] = \mathrm{neg}(n)$ for any PPT Inv.
- The real entropy of G is n

The generator

Definition 3

Given a function $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let G be the $(n+1)$-block generator $f(x)_1, \ldots, f(x)_n, x$.

Lemma 4

Assume that f is a OWF then G has accessible entropy at most $n - \log n$.

- Recall f is OWF if $\Pr_{x \leftarrow \{0,1\}^n; y=f(x)}\left[\mathsf{Inv}(y) \in f^{-1}(y)\right] = \mathrm{neg}(n)$ for any PPT Inv.
- The real entropy of G is n
- Hence, entropy gap is $\log n$

The generator

Definition 3

Given a function $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let G be the $(n+1)$-block generator $f(x)_1, \ldots, f(x)_n, x$.

Lemma 4

Assume that f is a OWF then G has accessible entropy at most $n - \log n$.

- Recall f is OWF if
  $\Pr_{x \leftarrow \{0,1\}^n; y = f(x)}\left[\mathrm{Inv}(y) \in f^{-1}(y)\right] = \mathrm{neg}(n)$ for any PPT Inv.
- The real entropy of G is n
- Hence, entropy gap is $\log n$
- Proof idea

Proving Lemma 4

Proving Lemma 4

Assume $\exists$ PPT $\widetilde{G}$ with $\Pr_{\mathbf{t} \leftarrow \widetilde{T}}\left[\mathrm{AccH}_{\mathsf{G},\widetilde{G}}(\mathbf{t}) > n - \log n\right] \geq \varepsilon = 1/\mathrm{poly}(n)$.

(recall $\widetilde{T} = (R_1, \widetilde{G}_1, \ldots, R_m, \widetilde{G}_m)$ is the coins and blocks of $\widetilde{G}$)

Proving Lemma 4

Assume $\exists$ PPT $\widetilde{G}$ with $\Pr_{\mathbf{t}\leftarrow\widetilde{T}}\left[\mathrm{AccH}_{\mathsf{G},\widetilde{G}}(\mathbf{t}) > n - \log n\right] \geq \varepsilon = 1/\mathrm{poly}(n)$.

(recall $\widetilde{T} = (R_1, \widetilde{G}_1, \ldots, R_m, \widetilde{G}_m)$ is the coins and blocks of $\widetilde{G}$)

Algorithm 5 ($\mathsf{Inv}(z)$)

1. For $i = 1$ to n, do the following for n^2/ε times:
   - **1.1** Sample r_i uniformly at random and let g_i be the i'th output block of $\widetilde{G}(r_1, \ldots, r_i)$.
   - **1.2** If $g_i = z_i$, move to next value of i.
   - **1.3** Abort, if the maximal number of attempts is reached.
2. Finish the execution of $\widetilde{G}(r_1, \ldots, r_{n+1})$, and output its $(n+1)$ output block.

Proving Lemma 4

Assume $\exists$ PPT $\widetilde{G}$ with $\Pr_{\mathbf{t}\leftarrow\widetilde{T}}\left[\mathrm{AccH}_{G,\widetilde{G}}(\mathbf{t}) > n - \log n\right] \geq \varepsilon = 1/\mathrm{poly}(n)$.
(recall $\widetilde{T} = (R_1, \widetilde{G}_1, \ldots, R_m, \widetilde{G}_m)$ is the coins and blocks of $\widetilde{G}$)

Algorithm 5 ($\mathrm{Inv}(z)$)

1. For $i = 1$ to n, do the following for n^2/ε times:
   1.1 Sample r_i uniformly at random and let g_i be the i'th output block of $\widetilde{G}(r_1, \ldots, r_i)$.
   1.2 If $g_i = z_i$, move to next value of i.
   1.3 Abort, if the maximal number of attempts is reached.
2. Finish the execution of $\widetilde{G}(r_1, \ldots, r_{n+1})$, and output its $(n+1)$ output block.

We finish the proof showing that

$$\Pr_{x\leftarrow\{0,1\}^n}\left[\mathrm{Inv}(f(x)) \in f^{-1}(f(x))\right] \geq \frac{\varepsilon}{4n}$$

Proving Lemma 4, cont.

Proving Lemma 4, cont.

Let $\mathcal{S} \subseteq \text{Supp}(\widetilde{T})$ denote the set of transcripts $\mathbf{t} = (r_1, g_1, \ldots, r_{n+1}, g_{n+1})$ with

1. $\text{AccH}_{G,\widetilde{G}}(\mathbf{t}) \geq n - \log n$, and
2. $H_{Y_i|\widetilde{G}_1,\ldots,\widetilde{G}_{i-1}}(g_i \mid g_1, \ldots, g_{i-1}) \leq \log(\frac{4n}{\varepsilon})$ for all $i \in [n]$.

Proving Lemma 4, cont.

Let $\mathcal{S} \subseteq \text{Supp}(\widetilde{T})$ denote the set of transcripts $\mathbf{t} = (r_1, g_1, \ldots, r_{n+1}, g_{n+1})$ with

1. $\text{AccH}_{G,\widetilde{G}}(\mathbf{t}) \geq n - \log n$, and
2. $H_{Y_i|\widetilde{G}_1,\ldots,\widetilde{G}_{i-1}}(g_i \mid g_1, \ldots, g_{i-1}) \leq \log(\frac{4n}{\varepsilon})$ for all $i \in [n]$.

Let $\mathcal{Z} := \{z \in \{0,1\}^n : \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$

Proving Lemma 4, cont.

Let $\mathcal{S} \subseteq \text{Supp}(\widetilde{T})$ denote the set of transcripts $\mathbf{t} = (r_1, g_1, \ldots, r_{n+1}, g_{n+1})$ with

1. $\text{AccH}_{G,\widetilde{G}}(\mathbf{t}) \geq n - \log n$, and
2. $H_{Y_i|\widetilde{G}_1,\ldots,\widetilde{G}_{i-1}}(g_i \mid g_1, \ldots, g_{i-1}) \leq \log(\frac{4n}{\varepsilon})$ for all $i \in [n]$.

Let $\mathcal{Z} := \{z \in \{0,1\}^n \colon \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$

For any $z \in \mathcal{Z}$:

$$\Pr\left[\text{Inv}(z) \in f^{-1}(z)\right] \geq 1 - n \cdot (1 - \tfrac{\varepsilon}{4n})^{n^2/\varepsilon} \geq 1 - O(n \cdot 2^{-n}) \geq \frac{1}{2}$$

Proving Lemma 4, cont.

Let $\mathcal{S} \subseteq \mathrm{Supp}(\widetilde{T})$ denote the set of transcripts $\mathbf{t} = (r_1, g_1, \ldots, r_{n+1}, g_{n+1})$ with

1. $\mathrm{AccH}_{G,\widetilde{G}}(\mathbf{t}) \geq n - \log n$, and
2. $H_{Y_i|\widetilde{G}_1,\ldots,\widetilde{G}_{i-1}}(g_i \mid g_1, \ldots, g_{i-1}) \leq \log(\frac{4n}{\varepsilon})$ for all $i \in [n]$.

Let $\mathcal{Z} := \{z \in \{0,1\}^n \colon \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$

For any $z \in \mathcal{Z}$:

$$\Pr\left[\mathrm{Inv}(z) \in f^{-1}(z)\right] \geq 1 - n \cdot (1 - \tfrac{\varepsilon}{4n})^{n^2/\varepsilon} \geq 1 - O(n \cdot 2^{-n}) \geq \frac{1}{2}$$

We complete the proof showing that

1. $\Pr_{\widetilde{T}}\left[\mathcal{S}\right] \geq \varepsilon/2$, and
2. $\Pr_{x \leftarrow \{0,1\}^n}\left[f(x) \in \mathcal{Z}\right] \geq \Pr_{\widetilde{T}}\left[\mathcal{S}\right]/n$

Proving Lemma 4, cont.

Let $\mathcal{S} \subseteq \text{Supp}(\widetilde{T})$ denote the set of transcripts $\mathbf{t} = (r_1, g_1, \ldots, r_{n+1}, g_{n+1})$ with

1. $\text{AccH}_{G,\widetilde{G}}(\mathbf{t}) \geq n - \log n$, and
2. $H_{Y_i|\widetilde{G}_1,\ldots,\widetilde{G}_{i-1}}(g_i \mid g_1, \ldots, g_{i-1}) \leq \log(\frac{4n}{\varepsilon})$ for all $i \in [n]$.

Let $\mathcal{Z} := \{z \in \{0,1\}^n : \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$

For any $z \in \mathcal{Z}$:

$$\Pr\left[\text{Inv}(z) \in f^{-1}(z)\right] \geq 1 - n \cdot (1 - \tfrac{\varepsilon}{4n})^{n^2/\varepsilon} \geq 1 - O(n \cdot 2^{-n}) \geq \frac{1}{2}$$

We complete the proof showing that

1. $\Pr_{\widetilde{T}}[\mathcal{S}] \geq \varepsilon/2$, and
2. $\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{Z}] \geq \Pr_{\widetilde{T}}[\mathcal{S}]/n$

Yielding that $\Pr_{x \leftarrow \{0,1\}^n}\left[\text{Inv}(f(x)) \in f^{-1}(f(x))\right] \geq \frac{\varepsilon}{4n}$.□

$\mathcal{S}$ is large

$\mathcal{S}$ is large

$$\Pr_{\widetilde{T}}[\mathcal{S}] \geq \Pr\left[\mathrm{AccH}_{G,\widetilde{G}}(T) \geq n - \log n\right]$$
$$- \Pr_{(g_1,\ldots,g_{n+1})\leftarrow(\widetilde{G}_1,\ldots,\widetilde{G}_{n+1})}\left[\exists i \in [n] \colon H_{\widetilde{G}_i|\widetilde{G}_1,\ldots,\widetilde{G}_{i-1}}(g_i \mid g_1,\ldots,g_{i-1}) > \log(\tfrac{4n}{\varepsilon})\right]$$

$\mathcal{S}$ is large

$$\Pr_{\widetilde{T}}[\mathcal{S}] \geq \Pr\left[\mathrm{AccH}_{G,\widetilde{G}}(T) \geq n - \log n\right]$$
$$- \Pr_{(g_1,\ldots,g_{n+1})\leftarrow(\widetilde{G}_1,\ldots,\widetilde{G}_{n+1})}\left[\exists i \in [n] \colon H_{\widetilde{G}_i|\widetilde{G}_1,\ldots,\widetilde{G}_{i-1}}(g_i \mid g_1,\ldots,g_{i-1}) > \log(\tfrac{4n}{\varepsilon})\right]$$
$$\geq \varepsilon - n \cdot 2 \cdot \tfrac{\varepsilon}{4n} = \varepsilon/2$$

$\mathcal{Z}$ is large

$\mathcal{Z}$ is large

For $t = (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathrm{Supp}(\widetilde{T})$ let

$$P(t) := \prod_{i=1}^{n+1} \Pr\left[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\right]$$

$\mathcal{Z}$ is large

For $t = (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathrm{Supp}(\widetilde{T})$ let

$$P(t) := \prod_{i=1}^{n+1} \Pr\left[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\right]$$

Compute

$$\Pr_{\widetilde{T}}[t] = \Pr[\widetilde{G}_1 = g_1] \cdot \Pr[R_1 = r_1 \mid \widetilde{G}_1 = g_1] \cdot \Pr[\widetilde{G}_2 = g_2 | R_1 = r_1] \cdot \Pr[R_2 = r_2 \mid \widetilde{G}_2 = g_2] \cdots \tag{1}$$

$\mathcal{Z}$ is large

For $t = (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathrm{Supp}(\widetilde{T})$ let

$$P(t) := \prod_{i=1}^{n+1} \Pr\left[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\right]$$

Compute

$$\begin{aligned}\Pr_{\widetilde{T}}[t] &= \Pr[\widetilde{G}_1 = g_1] \cdot \Pr[R_1 = r_1 \mid \widetilde{G}_1 = g_1] \\ &\quad \cdot \Pr[\widetilde{G}_2 = g_2 | R_1 = r_1] \cdot \Pr[R_2 = r_2 \mid \widetilde{G}_2 = g_2] \cdots \\ &= 2^{-\sum_{i=1}^{m} H_{\widetilde{G}_i | R_1,\ldots,R_{i-1}}(g_i | r_1,\ldots,r_{i-1})} \cdot P(t)\end{aligned} \tag{1}$$

$\mathcal{Z}$ is large

For $t = (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \operatorname{Supp}(\widetilde{T})$ let

$$P(t) := \prod_{i=1}^{n+1} \Pr\left[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\right]$$

Compute

$$\begin{aligned}\Pr_{\widetilde{T}}[t] &= \Pr[\widetilde{G}_1 = g_1] \cdot \Pr[R_1 = r_1 \mid \widetilde{G}_1 = g_1] \\ &\quad \cdot \Pr[\widetilde{G}_2 = g_2 | R_1 = r_1] \cdot \Pr[R_2 = r_2 \mid \widetilde{G}_2 = g_2] \cdots \\ &= 2^{-\sum_{i=1}^{m} H_{\widetilde{G}_i | R_1, \ldots, R_{i-1}}(g_i | r_1, \ldots, r_{i-1})} \cdot P(t) \\ &= 2^{-\operatorname{AccH}_{G,\widetilde{G}}(t)} \cdot P(t)\end{aligned} \tag{1}$$

$\mathcal{Z}$ is large, cont.

$\mathcal{Z}$ is large, cont.

For $t = (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathrm{Supp}(\widetilde{T})$.

$$P(t) = \prod_{i=1}^{n+1} \Pr\left[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\right]$$

$\mathcal{Z}$ is large, cont.

For $t = (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathrm{Supp}(\widetilde{T})$.

$$
\begin{aligned}
P(t) &= \prod_{i=1}^{n+1} \Pr\Big[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\Big] \\
&= \prod_{i=1}^{n+1} \Pr\Big[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\Big] \cdot \Pr[\widetilde{G}_i = g_i | \widetilde{G}_{n+1} = g_{n+1}]
\end{aligned}
$$

$\mathcal{Z}$ is large, cont.

For $t = (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathrm{Supp}(\widetilde{T})$.

$$
\begin{aligned}
P(t) &= \prod_{i=1}^{n+1} \Pr\Big[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\Big] \\
&= \prod_{i=1}^{n+1} \Pr\Big[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\Big] \cdot \Pr[\widetilde{G}_i = g_i | \widetilde{G}_{n+1} = g_{n+1}] \\
&= \prod_{i=1}^{n+1} \Pr\Big[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i, \widetilde{G}_{n+1}) = (r_{1,\ldots,i-1}, g_i, g_{n+1})\Big] \\
&\quad \cdot \Pr[\widetilde{G}_i = g_i | \widetilde{G}_{n+1} = g_{n+1}]
\end{aligned}
$$

$\mathcal{Z}$ is large, cont.

For $t = (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathrm{Supp}(\widetilde{T})$.

$$\begin{aligned}
P(t) &= \prod_{i=1}^{n+1} \Pr\left[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\right] \\
&= \prod_{i=1}^{n+1} \Pr\left[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i) = (r_{1,\ldots,i-1}, g_i)\right] \cdot \Pr[\widetilde{G}_i = g_i | \widetilde{G}_{n+1} = g_{n+1}] \\
&= \prod_{i=1}^{n+1} \Pr\left[R_i = r_i \mid (R_{1,\ldots,i-1}, \widetilde{G}_i, \widetilde{G}_{n+1}) = (r_{1,\ldots,i-1}, g_i, g_{n+1})\right] \\
&\quad \cdot \Pr[\widetilde{G}_i = g_i | \widetilde{G}_{n+1} = g_{n+1}] \\
&= \Pr_{\widetilde{T}}\left[t | \widetilde{G}_{n+1} = g_{n+1}\right]
\end{aligned}$$

$\mathcal{Z}$ is large, cont..

$\mathcal{Z}$ is large, cont..

- Recall, $\mathcal{Z} = \{z \in \{0,1\}^n \colon \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$

$\mathcal{Z}$ is large, cont..

- Recall, $\mathcal{Z} = \{z \in \{0,1\}^n : \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$
- By definition $2^{-\mathsf{AccH}_{G,\widetilde{G}}(\mathbf{t})} \leq n \cdot 2^{-n}$, for any $\mathbf{t} \in \mathcal{S}$

$\mathcal{Z}$ is large, cont..

- Recall, $\mathcal{Z} = \{z \in \{0,1\}^n \colon \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$
- By definition $2^{-\operatorname{AccH}_{G,\widetilde{G}}(\mathbf{t})} \leq n \cdot 2^{-n}$, for any $\mathbf{t} \in \mathcal{S}$
- We saw $\Pr_{\widetilde{T}}[\mathbf{t}] = 2^{-\operatorname{AccH}_{G,\widetilde{G}}(t)} \cdot \Pr_{\widetilde{T}}\left[t | \widetilde{G}_{n+1} = g_{n+1}\right]$ for any $\mathbf{t} \in \operatorname{Supp}(T)$.

$\mathcal{Z}$ is large, cont..

- Recall, $\mathcal{Z} = \{z \in \{0,1\}^n \colon \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$
- By definition $2^{-\operatorname{AccH}_{G,\widetilde{G}}(\mathbf{t})} \leq n \cdot 2^{-n}$, for any $\mathbf{t} \in \mathcal{S}$
- We saw $\Pr_{\widetilde{T}}[\mathbf{t}] = 2^{-\operatorname{AccH}_{G,\widetilde{G}}(t)} \cdot \Pr_{\widetilde{T}}\left[t \middle| \widetilde{G}_{n+1} = g_{n+1}\right]$ for any $\mathbf{t} \in \operatorname{Supp}(T)$.

$\mathcal{Z}$ is large, cont..

- Recall, $\mathcal{Z} = \{z \in \{0,1\}^n \colon \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$
- By definition $2^{-\operatorname{AccH}_{G,\widetilde{G}}(\mathbf{t})} \le n \cdot 2^{-n}$, for any $\mathbf{t} \in \mathcal{S}$
- We saw $\Pr_{\widetilde{T}}[\mathbf{t}] = 2^{-\operatorname{AccH}_{G,\widetilde{G}}(t)} \cdot \Pr_{\widetilde{T}}\left[t | \widetilde{G}_{n+1} = g_{n+1}\right]$ for any $\mathbf{t} \in \operatorname{Supp}(T)$.

Hence

$$\Pr_{\widetilde{T}}[\mathcal{S}] \le n \cdot 2^{-n} \cdot \sum_{\mathbf{t} \in \mathcal{S}} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = g_{n+1}\right]$$

$\mathcal{Z}$ is large, cont..

- Recall, $\mathcal{Z} = \{z \in \{0,1\}^n \colon \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$
- By definition $2^{-\operatorname{AccH}_{G,\widetilde{G}}(\mathbf{t})} \leq n \cdot 2^{-n}$, for any $\mathbf{t} \in \mathcal{S}$
- We saw $\Pr_{\widetilde{T}}[\mathbf{t}] = 2^{-\operatorname{AccH}_{G,\widetilde{G}}(t)} \cdot \Pr_{\widetilde{T}}\left[t | \widetilde{G}_{n+1} = g_{n+1}\right]$ for any $\mathbf{t} \in \operatorname{Supp}(T)$.

Hence

$$\begin{aligned}
\Pr_{\widetilde{T}}[\mathcal{S}] &\leq n \cdot 2^{-n} \cdot \sum_{\mathbf{t} \in \mathcal{S}} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = g_{n+1}\right] \\
&= n \cdot 2^{-n} \cdot \sum_{z \in \mathcal{Z}} \sum_{\mathbf{t} = (\ldots, g_{n+1}) \in \mathcal{S} : f(g_{n+1}) = z} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = g_{n+1}\right]
\end{aligned}$$

$\mathcal{Z}$ is large, cont..

- Recall, $\mathcal{Z} = \{z \in \{0,1\}^n \colon \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$
- By definition $2^{-\mathsf{AccH}_{G,\widetilde{G}}(\mathbf{t})} \leq n \cdot 2^{-n}$, for any $\mathbf{t} \in \mathcal{S}$
- We saw $\Pr_{\widetilde{T}}[\mathbf{t}] = 2^{-\mathsf{AccH}_{G,\widetilde{G}}(t)} \cdot \Pr_{\widetilde{T}}\left[t | \widetilde{G}_{n+1} = g_{n+1}\right]$ for any $\mathbf{t} \in \mathrm{Supp}(T)$.

Hence

$$\begin{aligned}
\Pr_{\widetilde{T}}[\mathcal{S}] &\leq n \cdot 2^{-n} \cdot \sum_{\mathbf{t} \in \mathcal{S}} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = g_{n+1}\right] \\
&= n \cdot 2^{-n} \cdot \sum_{z \in \mathcal{Z}} \sum_{\mathbf{t} = (\ldots, g_{n+1}) \in \mathcal{S} : f(g_{n+1}) = z} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = g_{n+1}\right] \\
&= n \cdot 2^{-n} \cdot \sum_{z \in \mathcal{Z}} \sum_{y \in f^{-1}(z)} \sum_{\mathbf{t} = (\ldots, y)} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = y\right]
\end{aligned}$$

$\mathcal{Z}$ is large, cont..

- Recall, $\mathcal{Z} = \{z \in \{0,1\}^n \colon \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$
- By definition $2^{-\operatorname{AccH}_{G,\widetilde{G}}(\mathbf{t})} \leq n \cdot 2^{-n}$, for any $\mathbf{t} \in \mathcal{S}$
- We saw $\Pr_{\widetilde{T}}[\mathbf{t}] = 2^{-\operatorname{AccH}_{G,\widetilde{G}}(t)} \cdot \Pr_{\widetilde{T}}\left[t | \widetilde{G}_{n+1} = g_{n+1}\right]$ for any $\mathbf{t} \in \operatorname{Supp}(T)$.

Hence

$$\begin{aligned}
\Pr_{\widetilde{T}}[\mathcal{S}] &\leq n \cdot 2^{-n} \cdot \sum_{\mathbf{t} \in \mathcal{S}} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = g_{n+1}\right] \\
&= n \cdot 2^{-n} \cdot \sum_{z \in \mathcal{Z}} \sum_{\mathbf{t} = (\ldots, g_{n+1}) \in \mathcal{S} : f(g_{n+1}) = z} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = g_{n+1}\right] \\
&= n \cdot 2^{-n} \cdot \sum_{z \in \mathcal{Z}} \sum_{y \in f^{-1}(z)} \sum_{\mathbf{t} = (\ldots, y)} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = y\right] \\
&\leq n \cdot 2^{-n} \cdot \sum_{z \in \mathcal{Z}} \left| f^{-1}(z) \right|
\end{aligned}$$

$\mathcal{Z}$ is large, cont..

- Recall, $\mathcal{Z} = \{z \in \{0,1\}^n \colon \exists (r_1, g_1, \ldots, r_{n+1}, g_{n+1}) \in \mathcal{S} \text{ s.t. } f(g_{n+1}) = z\}$
- By definition $2^{-\mathsf{AccH}_{G,\widetilde{G}}(\mathbf{t})} \leq n \cdot 2^{-n}$, for any $\mathbf{t} \in \mathcal{S}$
- We saw $\Pr_{\widetilde{T}}[\mathbf{t}] = 2^{-\mathsf{AccH}_{G,\widetilde{G}}(t)} \cdot \Pr_{\widetilde{T}}\left[t | \widetilde{G}_{n+1} = g_{n+1}\right]$ for any $\mathbf{t} \in \mathsf{Supp}(T)$.

Hence

$$\begin{aligned}
\Pr_{\widetilde{T}}[\mathcal{S}] &\leq n \cdot 2^{-n} \cdot \sum_{\mathbf{t} \in \mathcal{S}} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = g_{n+1}\right] \\
&= n \cdot 2^{-n} \cdot \sum_{z \in \mathcal{Z}} \sum_{\mathbf{t} = (\ldots, g_{n+1}) \in \mathcal{S} : f(g_{n+1}) = z} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = g_{n+1}\right] \\
&= n \cdot 2^{-n} \cdot \sum_{z \in \mathcal{Z}} \sum_{y \in f^{-1}(z)} \sum_{\mathbf{t} = (\ldots, y)} \Pr_{\widetilde{T}}\left[\mathbf{t} | \widetilde{G}_{n+1} = y\right] \\
&\leq n \cdot 2^{-n} \cdot \sum_{z \in \mathcal{Z}} \left| f^{-1}(z) \right| \\
&= n \cdot \Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{Z}]. \square
\end{aligned}$$

Section 5

SHC from Inaccessible Entropy

High-level description

High-level description

- Entropy equalization + gap amplification to get generator that has the same min-entropy in each block and whose accessible entropy is n-bit smaller than the sum of the min entropies.

High-level description

- Entropy equalization + gap amplification to get generator that has the same min-entropy in each block and whose accessible entropy is n-bit smaller than the sum of the min entropies.
- Use universal hashing to get a "generator" with zero accessible entropy block

High-level description

- Entropy equalization + gap amplification to get generator that has the same min-entropy in each block and whose accessible entropy is n-bit smaller than the sum of the min entropies.
- Use universal hashing to get a "generator" with zero accessible entropy block
- Use target-collision-resistant hash family (a non-interactive cryptographic tool implied by OWF) to get weakly binding SHC

High-level description

- Entropy equalization + gap amplification to get generator that has the same min-entropy in each block and whose accessible entropy is n-bit smaller than the sum of the min entropies.
- Use universal hashing to get a "generator" with zero accessible entropy block
- Use target-collision-resistant hash family (a non-interactive cryptographic tool implied by OWF) to get weakly binding SHC
- Amplify the above into full-fledged SHC

Hashing protocol

Let $\mathcal{T} \subseteq \{0,1\}^{\ell}$ be 2^{k}-size set.
Let $\mathcal{H}^{1}$ be ℓ-wise independent family mapping ℓ-bit strings to k-bit strings
Let $\mathcal{H}^{2}$ be 2-universal family mapping ℓ-length strings to n-bit strings

Hashing protocol

Let $\mathcal{T} \subseteq \{0,1\}^\ell$ be 2^k-size set.
Let $\mathcal{H}^1$ be ℓ-wise independent family mapping ℓ-bit strings to k-bit strings
Let $\mathcal{H}^2$ be 2-universal family mapping ℓ-length strings to n-bit strings

Protocol 6 ((S, R))

1. S selects $x \in \mathcal{T}$
2. R sends $h^1 \leftarrow \mathcal{H}^1$ to S
3. S sends $y^1 = h^1(x)$ to R
4. R sends $h^2 \leftarrow \mathcal{H}^2$ to S
5. S sends $y^2 = h^2(x)$ to R

Hashing protocol

Let $\mathcal{T} \subseteq \{0,1\}^\ell$ be 2^k-size set.
Let $\mathcal{H}^1$ be ℓ-wise independent family mapping ℓ-bit strings to k-bit strings
Let $\mathcal{H}^2$ be 2-universal family mapping ℓ-length strings to n-bit strings

Protocol 6 ((S, R))

1. S selects $x \in \mathcal{T}$
2. R sends $h^1 \leftarrow \mathcal{H}^1$ to S
3. S sends $y^1 = h^1(x)$ to R
4. R sends $h^2 \leftarrow \mathcal{H}^2$ to S
5. S sends $y^2 = h^2(x)$ to R

Let $\widetilde{\mathsf{S}}$ be an arbitrary algorithm and let Y^1, Y^2, H^1, H^2 be value of y^1, y^2, h^1, h^2 in a random execution of $(\widetilde{\mathsf{S}}, \mathsf{R})$.

Hashing protocol

Let $\mathcal{T} \subseteq \{0,1\}^\ell$ be 2^k-size set.
Let $\mathcal{H}^1$ be ℓ-wise independent family mapping ℓ-bit strings to k-bit strings
Let $\mathcal{H}^2$ be 2-universal family mapping ℓ-length strings to n-bit strings

Protocol 6 ((S, R))

1. S selects $x \in \mathcal{T}$
2. R sends $h^1 \leftarrow \mathcal{H}^1$ to S
3. S sends $y^1 = h^1(x)$ to R
4. R sends $h^2 \leftarrow \mathcal{H}^2$ to S
5. S sends $y^2 = h^2(x)$ to R

Let $\widetilde{\mathsf{S}}$ be an arbitrary algorithm and let Y^1, Y^2, H^1, H^2 be value of y^1, y^2, h^1, h^2 in a random execution of $(\widetilde{\mathsf{S}}, \mathsf{R})$.

Claim 7

$\Pr\left[\exists x \neq x' \in \mathcal{T} : H^1(x) = H^1(x') = Y^1 \wedge H^2(x) = H^3(x') = Y^3\right] \in 2^{-\Omega(n)}$.

Hashing protocol

Let $\mathcal{T} \subseteq \{0,1\}^\ell$ be 2^k-size set.
Let $\mathcal{H}^1$ be ℓ-wise independent family mapping ℓ-bit strings to k-bit strings
Let $\mathcal{H}^2$ be 2-universal family mapping ℓ-length strings to n-bit strings

Protocol 6 ((S, R))

1. S selects $x \in \mathcal{T}$
2. R sends $h^1 \leftarrow \mathcal{H}^1$ to S
3. S sends $y^1 = h^1(x)$ to R
4. R sends $h^2 \leftarrow \mathcal{H}^2$ to S
5. S sends $y^2 = h^2(x)$ to R

Let $\widetilde{\mathsf{S}}$ be an arbitrary algorithm and let Y^1, Y^2, H^1, H^2 be value of y^1, y^2, h^1, h^2 in a random execution of $(\widetilde{\mathsf{S}}, \mathsf{R})$.

Claim 7

$\Pr\left[\exists x \neq x' \in \mathcal{T} \colon H^1(x) = H^1(x') = Y^1 \land H^2(x) = H^3(x') = Y^3\right] \in 2^{-\Omega(n)}$.

Proof: ?

Hashing protocol

Let $\mathcal{T} \subseteq \{0,1\}^\ell$ be 2^k-size set.
Let $\mathcal{H}^1$ be ℓ-wise independent family mapping ℓ-bit strings to k-bit strings
Let $\mathcal{H}^2$ be 2-universal family mapping ℓ-length strings to n-bit strings

Protocol 6 ((S,R))

1. S selects $x \in \mathcal{T}$
2. R sends $h^1 \leftarrow \mathcal{H}^1$ to S
3. S sends $y^1 = h^1(x)$ to R
4. R sends $h^2 \leftarrow \mathcal{H}^2$ to S
5. S sends $y^2 = h^2(x)$ to R

Let $\widetilde{\mathsf{S}}$ be an arbitrary algorithm and let Y^1, Y^2, H^1, H^2 be value of y^1, y^2, h^1, h^2 in a random execution of $(\widetilde{\mathsf{S}},\mathsf{R})$.

Claim 7

$\Pr\left[\exists x \neq x' \in \mathcal{T} \colon H^1(x) = H^1(x') = Y^1 \wedge H^2(x) = H^3(x') = Y^3\right] \in 2^{-\Omega(n)}$.

Proof: ? Can we do it in a single round?

"Generator" with zero accessible entropy block

“Generator” with zero accessible entropy block

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}^1$ be ℓ-wise function family mapping ℓ-bit strings of k-bit strings. Let $\mathcal{H}^2$ be 2-universal function family mapping ℓ-bit strings to n-bit strings.

"Generator" with zero accessible entropy block

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}^1$ be ℓ-wise function family mapping ℓ-bit strings of k-bit strings. Let $\mathcal{H}^2$ be 2-universal function family mapping ℓ-bit strings to n-bit strings.

Protocol 8 ($G' = (\mathsf{S}, \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$

For $i = 1$ to m:

1. R sends $h_i^1 \leftarrow \mathcal{H}^1$ to S
2. S sends $y_i^1 = h_i^1(G(x)_i)$ to R
3. R sends $h_i^2 \leftarrow \mathcal{H}^2$ to S
4. S sends $y_i^2 = h_i^2(G(x)_i)$ to R
5. S sends $g_i = G(x)_i$ to R

"Generator" with zero accessible entropy block

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}^1$ be ℓ-wise function family mapping ℓ-bit strings of k-bit strings. Let $\mathcal{H}^2$ be 2-universal function family mapping ℓ-bit strings to n-bit strings.

Protocol 8 ($G' = (\mathsf{S}, \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$

For $i = 1$ to m:

1. R sends $h_i^1 \leftarrow \mathcal{H}^1$ to S
2. S sends $y_i^1 = h_i^1(G(x)_i)$ to R
3. R sends $h_i^2 \leftarrow \mathcal{H}^2$ to S
4. S sends $y_i^2 = h_i^2(G(x)_i)$ to R
5. S sends $g_i = G(x)_i$ to R

- We view G' as an m-block "interactive generator" (the blocks are $g_1, \ldots, g_m$).

"Generator" with zero accessible entropy block

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}^1$ be ℓ-wise function family mapping ℓ-bit strings of k-bit strings. Let $\mathcal{H}^2$ be 2-universal function family mapping ℓ-bit strings to n-bit strings.

Protocol 8 ($G' = (\mathsf{S}, \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$

For $i = 1$ to m:

1. R sends $h_i^1 \leftarrow \mathcal{H}^1$ to S
2. S sends $y_i^1 = h_i^1(G(x)_i)$ to R
3. R sends $h_i^2 \leftarrow \mathcal{H}^2$ to S
4. S sends $y_i^2 = h_i^2(G(x)_i)$ to R
5. S sends $g_i = G(x)_i$ to R

- We view G' as an m-block "interactive generator" (the blocks are $g_1, \ldots, g_m$).
- Assume the blocks of G has real min-entropy $(k + n + t)$, then the blocks of G' has real min-entropy roughly t

"Generator" with zero accessible entropy block

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}^1$ be ℓ-wise function family mapping ℓ-bit strings of k-bit strings. Let $\mathcal{H}^2$ be 2-universal function family mapping ℓ-bit strings to n-bit strings.

Protocol 8 ($G' = (\mathsf{S}, \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$

For $i = 1$ to m:

1. R sends $h_i^1 \leftarrow \mathcal{H}^1$ to S
2. S sends $y_i^1 = h_i^1(G(x)_i)$ to R
3. R sends $h_i^2 \leftarrow \mathcal{H}^2$ to S
4. S sends $y_i^2 = h_i^2(G(x)_i)$ to R
5. S sends $g_i = G(x)_i$ to R

- We view G' as an m-block "interactive generator" (the blocks are $g_1, \ldots, g_m$).
- Assume the blocks of G has real min-entropy $(k + n + t)$, then the blocks of G' has real min-entropy roughly t
- Assume G has accessible entropy mk, then w.p. $1 - negl(n)$ in an execution of G' exists block with accessible entropy 0:

"Generator" with zero accessible entropy block

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}^1$ be ℓ-wise function family mapping ℓ-bit strings of k-bit strings. Let $\mathcal{H}^2$ be 2-universal function family mapping ℓ-bit strings to n-bit strings.

Protocol 8 ($G' = (\mathsf{S}, \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$

For $i = 1$ to m:

1. R sends $h_i^1 \leftarrow \mathcal{H}^1$ to S
2. S sends $y_i^1 = h_i^1(G(x)_i)$ to R
3. R sends $h_i^2 \leftarrow \mathcal{H}^2$ to S
4. S sends $y_i^2 = h_i^2(G(x)_i)$ to R
5. S sends $g_i = G(x)_i$ to R

- We view G' as an m-block "interactive generator" (the blocks are $g_1, \ldots, g_m$).
- Assume the blocks of G has real min-entropy $(k + n + t)$, then the blocks of G' has real min-entropy roughly t
- Assume G has accessible entropy mk, then w.p. $1 - negl(n)$ in an execution of G' exists block with accessible entropy 0:

"Generator" with zero accessible entropy block

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}^1$ be ℓ-wise function family mapping ℓ-bit strings of k-bit strings. Let $\mathcal{H}^2$ be 2-universal function family mapping ℓ-bit strings to n-bit strings.

Protocol 8 ($G' = (\mathrm{S}, \mathrm{R})$)

S sets $x \leftarrow \{0,1\}^s$

For $i = 1$ to m:

1. R sends $h_i^1 \leftarrow \mathcal{H}^1$ to S
2. S sends $y_i^1 = h_i^1(G(x)_i)$ to R
3. R sends $h_i^2 \leftarrow \mathcal{H}^2$ to S
4. S sends $y_i^2 = h_i^2(G(x)_i)$ to R
5. S sends $g_i = G(x)_i$ to R

- We view G' as an m-block "interactive generator" (the blocks are $g_1, \ldots, g_m$).
- Assume the blocks of G has real min-entropy $(k + n + t)$, then the blocks of G' has real min-entropy roughly t
- Assume G has accessible entropy mk, then w.p. $1 - negl(n)$ in an execution of G' exists block with accessible entropy 0: $H_{\widetilde{G}_i|R_1,\ldots,R_{i-1},H_1,\ldots,H_i,Y_i}(g_i|r_1,\ldots,r_{i-1},(h_1^1,h_1^2),\ldots,(h_i^1,h_i^2),(y_i^1,y_i^2)) = 0$), where H_i/Y_i are the values of $(h_i^1,h_i^2)/(y_i^1,y_i^2)$ in random execution of $\widetilde{G}$.

Target collision-resistant functions

Target collision-resistant functions

Definition 9 (target collision-resistant functions (TCR))

A function family $\mathcal{H} = \{\mathcal{H}_n\}$ is target collision resistant, if

$$\Pr_{(x,a)\leftarrow \mathsf{A}_1(1^n);h\leftarrow\mathcal{H}_n;x'\leftarrow \mathsf{A}_2(a,h)}\left[x \neq x' \wedge h(x) = h(x')\right] = \operatorname{neg}(n)$$

for any pair of PPT's $\mathsf{A}_1, \mathsf{A}_2$.

Target collision-resistant functions

Definition 9 (target collision-resistant functions (TCR))

A function family $\mathcal{H} = \{\mathcal{H}_n\}$ is target collision resistant, if

$$\Pr_{(x,a)\leftarrow \mathsf{A}_1(1^n);h\leftarrow \mathcal{H}_n;x'\leftarrow \mathsf{A}_2(a,h)}\left[x \neq x' \wedge h(x) = h(x')\right] = \operatorname{neg}(n)$$

for any pair of PPT's $\mathsf{A}_1, \mathsf{A}_2$.

Relaxed variant of collision resistant.

Target collision-resistant functions

Definition 9 (target collision-resistant functions (TCR))

A function family $\mathcal{H} = \{\mathcal{H}_n\}$ is target collision resistant, if

$$\Pr_{(x,a)\leftarrow \mathsf{A}_1(1^n); h\leftarrow \mathcal{H}_n; x'\leftarrow \mathsf{A}_2(a,h)}\left[x \neq x' \wedge h(x) = h(x')\right] = \operatorname{neg}(n)$$

for any pair of PPT's $\mathsf{A}_1, \mathsf{A}_2$.

Relaxed variant of collision resistant.

Theorem 10

OWFs imply efficient compressing TCRs.

Weakly binding SHC

Weakly binding SHC

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}$ be a TCR family mapping strings of length ℓ to string of length k. Let $\mathcal{G}$ be 2-universal Boolean function family over strings of length ℓ.

Weakly binding SHC

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}$ be a TCR family mapping strings of length ℓ to string of length k. Let $\mathcal{G}$ be 2-universal Boolean function family over strings of length ℓ.

Protocol 11 ($\mathsf{Com} = (\mathsf{S}(\sigma), \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$ and R sets $i^* \leftarrow [m]$

For $i = 1$ to m:

1. R sends $h_i \leftarrow \mathcal{H}$ to S
2. S sends $y_i = h_i(G(x)_i)$ to R
3. If $i = i^*$:
   - **3.1** R sends $g \leftarrow \mathcal{G}$ to S
   - **3.2** S sends $g(G(x)_i) \oplus \sigma$ to R
   - **3.3** Parties **stop** the execution.

Weakly binding SHC

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}$ be a TCR family mapping strings of length ℓ to string of length k. Let $\mathcal{G}$ be 2-universal Boolean function family over strings of length ℓ.

Protocol 11 ($\mathsf{Com} = (\mathsf{S}(\sigma), \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$ and R sets $i^* \leftarrow [m]$

For $i = 1$ to m:

1. R sends $h_i \leftarrow \mathcal{H}$ to S
2. S sends $y_i = h_i(G(x)_i)$ to R
3. If $i = i^*$:
   - **3.1** R sends $g \leftarrow \mathcal{G}$ to S
   - **3.2** S sends $g(G(x)_i) \oplus \sigma$ to R
   - **3.3** Parties **stop** the execution.

- Assume the blocks of G has real min entropy $(k + n)$, then Com is statistically hiding

Weakly binding SHC

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}$ be a TCR family mapping strings of length ℓ to string of length k. Let $\mathcal{G}$ be 2-universal Boolean function family over strings of length ℓ.

Protocol 11 ($\mathsf{Com} = (\mathsf{S}(\sigma), \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$ and R sets $i^* \leftarrow [m]$

For $i = 1$ to m:

1. R sends $h_i \leftarrow \mathcal{H}$ to S
2. S sends $y_i = h_i(G(x)_i)$ to R
3. If $i = i^*$:
   - 3.1 R sends $g \leftarrow \mathcal{G}$ to S
   - 3.2 S sends $g(G(x)_i) \oplus \sigma$ to R
   - 3.3 Parties **stop** the execution.

- Assume the blocks of G has real min entropy $(k+n)$, then Com is statistically hiding
- Assume G has a zero entropy block, then Com is $\frac{1}{m}$ binding.

Weakly binding SHC

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}$ be a TCR family mapping strings of length ℓ to string of length k. Let $\mathcal{G}$ be 2-universal Boolean function family over strings of length ℓ.

Protocol 11 ($\mathsf{Com} = (\mathsf{S}(\sigma), \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$ and R sets $i^* \leftarrow [m]$

For $i = 1$ to m:

1. R sends $h_i \leftarrow \mathcal{H}$ to S
2. S sends $y_i = h_i(G(x)_i)$ to R
3. If $i = i^*$:
   - 3.1 R sends $g \leftarrow \mathcal{G}$ to S
   - 3.2 S sends $g(G(x)_i) \oplus \sigma$ to R
   - 3.3 Parties **stop** the execution.

- Assume the blocks of G has real min entropy $(k + n)$, then Com is statistically hiding
- Assume G has a zero entropy block, then Com is $\frac{1}{m}$ binding.

Weakly binding SHC

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}$ be a TCR family mapping strings of length ℓ to string of length k. Let $\mathcal{G}$ be 2-universal Boolean function family over strings of length ℓ.

Protocol 11 (Com = (S(σ), R))

S sets $x \leftarrow \{0,1\}^s$ and R sets $i^* \leftarrow [m]$
For $i = 1$ to m:

1. R sends $h_i \leftarrow \mathcal{H}$ to S
2. S sends $y_i = h_i(G(x)_i)$ to R
3. If $i = i^*$:
   - **3.1** R sends $g \leftarrow \mathcal{G}$ to S
   - **3.2** S sends $g(G(x)_i) \oplus \sigma$ to R
   - **3.3** Parties **stop** the execution.

- Assume the blocks of G has real min entropy $(k+n)$, then Com is statistically hiding
- Assume G has a zero entropy block, then Com is $\frac{1}{m}$ binding. Proof:

Weakly binding SHC

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}$ be a TCR family mapping strings of length ℓ to string of length k. Let $\mathcal{G}$ be 2-universal Boolean function family over strings of length ℓ.

Protocol 11 ($\mathsf{Com} = (\mathsf{S}(\sigma), \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$ and R sets $i^* \leftarrow [m]$
For $i = 1$ to m:

1. R sends $h_i \leftarrow \mathcal{H}$ to S
2. S sends $y_i = h_i(G(x)_i)$ to R
3. If $i = i^*$:
   - 3.1 R sends $g \leftarrow \mathcal{G}$ to S
   - 3.2 S sends $g(G(x)_i) \oplus \sigma$ to R
   - 3.3 Parties **stop** the execution.

- Assume the blocks of G has real min entropy $(k + n)$, then Com is statistically hiding
- Assume G has a zero entropy block, then Com is $\frac{1}{m}$ binding. Proof:
  1. For some $i \in [m]$, cheating $\widetilde{\mathsf{S}}$ must send hash of zero-entropy block.

Weakly binding SHC

Let G be m-block generator of block size ℓ and input length s. Let $\mathcal{H}$ be a TCR family mapping strings of length ℓ to string of length k. Let $\mathcal{G}$ be 2-universal Boolean function family over strings of length ℓ.

Protocol 11 ($\mathsf{Com} = (\mathsf{S}(\sigma), \mathsf{R})$)

S sets $x \leftarrow \{0,1\}^s$ and R sets $i^* \leftarrow [m]$

For $i = 1$ to m:

1. R sends $h_i \leftarrow \mathcal{H}$ to S
2. S sends $y_i = h_i(G(x)_i)$ to R
3. If $i = i^*$:
   - 3.1 R sends $g \leftarrow \mathcal{G}$ to S
   - 3.2 S sends $g(G(x)_i) \oplus \sigma$ to R
   - 3.3 Parties **stop** the execution.

- Assume the blocks of G has real min entropy $(k+n)$, then Com is statistically hiding
- Assume G has a zero entropy block, then Com is $\frac{1}{m}$ binding. Proof:
  1. For some $i \in [m]$, cheating $\widetilde{\mathsf{S}}$ must send hash of zero-entropy block.
  2. If $i^* = i$, we have binding

Remarks

Remarks

- OWF over n bits implies $\Theta(n)$-round SHC

Remarks

- OWF over n bits implies $\Theta(n)$-round SHC
- Can be pushed to $\Theta(n/\log n)$ rounds

Remarks

- OWF over n bits implies $\Theta(n)$-round SHC
- Can be pushed to $\Theta(n/\log n)$ rounds
- Tight (at least for certain type of reductions)